BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

DIALOG(R)File 351:Derwent WPI

012600618 **Image available**
WPI Acc No: 1999-406722/199935
XRPX Acc No: N99-303371

Coding method that takes into account predetermined integer equal to or greater than 2, number greater than or equal to 1 of sequences of binary data representing physical quantity

Patent Assignee: CANON KK (CANO)
Inventor: LE DANTEC C; PIRET P
Number of Countries: 028 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 928071	A1	19990707	EP 98403283	A	19981223	199935	B
FR 2773287	A1	19990702	FR 9716669	A	19971230	199935	
JP 11298339	A	19991029	JP 99241	A	19990104	200003	
CN 1232323	A	19991020	CN 98125950	A	19981230	200009	
FR 2785742	A1	20000512	FR 9814084	A	19981109	200031	
KR 99063573	A	19990726	KR 9862589	A	19981230	200043	

Priority Applications (No Type Date): FR 9814084 A 19981109; FR 9716669 A 19971230
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 928071	A1	E	42	H03M-013/00	
Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI					
FR 2773287	A1			H03M-013/22	
JP 11298339	A		110	H03M-013/22	
CN 1232323	A			H03M-013/22	
FR 2785742	A1			H03M-013/23	
KR 99063573	A			H03M-007/00	

Abstract (Basic): EP 928071 A1

NOVELTY - Coding method takes into account preset integer equal or greater than 2, number, greater than or equal to 1, of binary data sequences of physical quantities. Each sequence has polynomial being multiple of preset polynomial and several binary data equal to product of any integer number M and integer NO, smallest integer so that given polynomial is divisible by each of the other polynomials.

USE - For providing a coding device, a decoding device and a method and a system for implementing them.

ADVANTAGE - Any error estimation by the corresponding decoder converges, and the case where convergence does not occur is therefore excluded by using this solution.

DESCRIPTION OF DRAWING(S) - The drawing shows schematically an operating flow diagram of the coder.

pp; 42 DwgNo 8/9

Title Terms: CODE; METHOD; ACCOUNT; PREDETERMINED; INTEGER; EQUAL; GREATER; NUMBER; GREATER; EQUAL; SEQUENCE; BINARY; DATA; REPRESENT; PHYSICAL; QUANTITY
Derwent Class: P86; T01; U21; U22; W01
International Patent Class (Main): H03M-007/00; H03M-013/00; H03M-013/22; H03M-013/23
International Patent Class (Additional): G06F-011/00; G06F-011/10; G10L-019/00; H04L-012/56; H04N-003/00; H04N-005/44
File Segment: EPI; EngPI
Manual Codes (EPI/S-X): T01-J04B2; U21-A06; U22-G01; W01-A01B2; W01-A01X

(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 773 287**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **97 16669**

(51) Int Cl⁶ : H 03 M 13/22, H 04 N 5/44, 3/00, H 04 L 12/56

(12) DEMANDE DE BREVET D'INVENTION A1

(22) Date de dépôt : 30.12.97.

(30) Priorité :

(43) Date de mise à la disposition du public de la demande : 02.07.99 Bulletin 99/26.

(56) Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

(60) Références à d'autres documents nationaux apparentés :

(71) Demandeur(s) : *CANON KABUSHIKI KAISHA* — JP.

(72) Inventeur(s) : LE DANTEC CLAUDE et PIRET PHILIPPE.

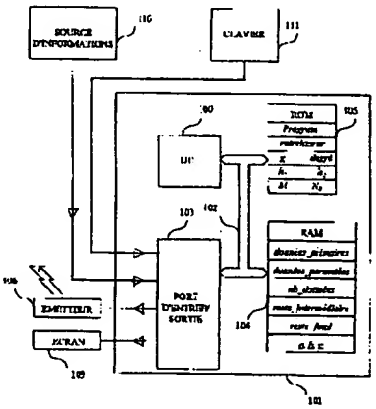
(73) Titulaire(s) :

(74) Mandataire(s) : RINUY SANTARELLI.

(54) ENTRELACEUR, DISPOSITIF DE CODAGE, PROCEDE DE PERMUTATION, PROCEDE DE CODAGE, DISPOSITIF ET PROCEDE DE DECODAGE ET SYSTEMES LES METTANT EN OEUVRE.

(57) L'invention expose des procédés et dispositifs de permutation (101) fournissant, à partir d'une séquence a de données binaires représentatives d'une grandeur physique, et divisible par une séquence g, une séquence permutée a*:

- a possède un nombre de données binaires égal au produit d'un nombre entier M quelconque par N0, plus petit entier tel que $X^{N0} - 1$ soit divisible par le polynôme g (x), et
- dans une représentation où les données binaires de la séquence a sont classées dans un tableau de N0 colonnes et de M lignes, la permutation considérée comporte un automorphisme du code cyclique binaire de longueur N0 et de polynôme générateur g (x), et des permutations travaillant uniquement sur des données d'une même colonne.



FR 2 773 287 - A1

La présente invention concerne un entrelaceur, un dispositif de codage, un procédé de permutation, un procédé de codage, un dispositif et un procédé de décodage et des systèmes les mettant en oeuvre.

Elle s'applique aussi bien au codage de données représentatives d'une grandeur physique, au codage de données sous forme de codes susceptibles de moduler une grandeur physique, au décodage de signaux modulés en données, qu'au décodage de données représentatives de grandeur physique. Ces données peuvent, par exemple, représenter des images, des sons, des données informatiques, des grandeurs électriques, des données mémorisées.

L'invention trouve une application dans le domaine des codes convolutifs. Lorsqu'on utilise ces derniers pour mettre en oeuvre un décodage itératif, ces codes sont fortement améliorés lorsque leurs codeurs contiennent un dispositif de permutation. Dans ce cas, ils sont habituellement appelés "turbocodes" et le décodeur itératif correspondant est appelé "turbodécodeur".

Sur ces sujets, des documents qui servent de référence sont, d'une part, l'article de MM. C. BERROU, A. GLAVIEUX et P. THITIMAJSHIMA intitulé "*Near Shannon limit error-correcting coding and decoding : turbocodes*" publiés avec les compte-rendus de la conférence "ICC'93", 1993, pages 1064 à 1070, et d'autre part, l'article de MM. C. BERROU et A. GLAVIEUX intitulé "*Near Optimum error-correcting coding and decoding : turbo-codes*" publié par IEEE

Transactions on Communication, Volume COM-44, pages 1261 à 1271, en octobre 1996.

Cependant, la formation des dispositifs de permutation est loin d'être parfaitement maîtrisée. En général ce dispositif utilise des matrices carrées ou rectangulaires dans lesquelles on écrit une ligne après l'autre et on lit une colonne après l'autre. Ces matrices sont généralement très grandes, par exemple de dimension 256 x 256.

Selon une autre méthode, dans un article intitulé "*Weight distributions for turbo-codes using random and nonrandom permutations*" publié par le Jet Propulsion Laboratory, avec "TDA Progress Report", numéro 42-122, le 15 août 1995, MM. DOLINAR et DIVSALAR considèrent les permutations qui, en numérotant les k positions d'information entre 0 et k-1, déplacent les informations binaires placées en une position i jusqu'à une position $e\ i + f$, pour des valeurs "bien choisies" de e et de f.

Dans ce document, ils ne donnent qu'un exemple où k est une puissance de 2. De plus, ils ne discutent pas l'influence réciproque possible du choix du dispositif de permutation et de celui des codeurs convolutifs élémentaires (2,1) à utiliser pour générer les séquences codées produites par le turbocodeur (3,1).

L'évaluation du turbocode correspondant consiste à simuler son utilisation sur un canal de transmission avec différentes valeurs de rapport signal/bruit et à mesurer la valeur minimum de ce rapport pour laquelle une valeur prédéterminée de probabilité d'erreur sur les valeurs binaires est atteinte.

Cependant, l'usage des simulations comme outil d'évaluation peut mener à quelques problèmes.

Considérons, par exemple, le dispositif de permutation avec k = 65536 = 256 x 256, évoqué plus haut, et choisissons une probabilité d'erreur prédéterminée égale à 10^{-5} pour simuler les performances d'un turbocode utilisant ce dispositif. En conséquence, le nombre moyen d'erreurs sur les valeurs

binaires par bloc de 256 x 256 sera proche de 1, mais nous ne saurons pas si chaque information binaire possède la même probabilité d'erreur. Cette probabilité d'erreur pourrait être assez élevée pour des valeurs binaires possédant une position "malheureuse" dans le dispositif de permutation et cette probabilité pourrait être beaucoup plus petite pour des positions plus "heureuses".

Une voie possible pour remédier à cette situation est de réaliser une conception harmonieuse et conjointe du dispositif de permutation et des deux codeurs convolutifs pour garantir une uniformité raisonnable du taux d'erreur sur les valeurs binaires après décodage, en fonction de la position de l'information binaire dans le dispositif de permutation.

Un autre problème concerne le manque d'outils algébriques pour spécifier les dispositifs de permutation. Il serait utile de disposer de moyens permettant de spécifier une sélection de dispositifs de permutation possédant des performances représentatives du jeu de tous les dispositifs de permutation.

L'invention concerne principalement la transmission d'information représentée par des séquences de symboles binaires :

$\underline{u} = (u_0, u_1, ..., u_{k-1})$,

appelées "séquences d'information", que l'on va coder en un triplet de séquences binaires,

$\underline{v} = (\underline{a}, \underline{b}, \underline{c})$,

chacune de ces séquences $\underline{a}$, $\underline{b}$ et $\underline{c}$, étant, à elle seule, représentative de la séquence $\underline{u}$.

Dans la suite de la description, on utilise indifféremment pour représenter une séquence $\underline{u}$, la forme $\underline{u} = (u_0, u_1, ..., u_{k-1})$, et la forme polynomiale associée :

$$u(x) = u_0 x^0 + u_1 x^1 + ... + u_{k-1} x^{k-1}.$$

Des notations équivalentes seront utilisées pour les séquences $\underline{a}$, $\underline{b}$ et $\underline{c}$. En utilisant cette seconde représentation, il est connu pour déterminer le triplet $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$:

- de choisir $a(x) = u(x)$;

- de choisir $b(x) = u(x).h_1(x) / g(x)$,

où $g(x)$ est un polynôme, par exemple $g(x) = 1 + x + x^3$, correspondant, selon la représentation en séquence, à la séquence (1, 1, 0, 1) ; et $h_1(x)$ est un polynôme, par exemple $h_1(x) = 1 + x + x^2 + x^3$, correspondant à la séquence (1, 1, 1, 1) ; et

- en appelant $\underline{a}^*$ une séquence formée par permutation des données binaires de la séquence $\underline{a}$, de choisir $c(x) = a^*(x).h_2(x) / g(x)$

où $h_2(x)$ est un polynôme, par exemple $h_2(x) = (1 + x^2 + x^3)$ correspondant à la séquence (1, 0, 1, 1).

Tout choix des polynômes $g(x)$, $h_1(x)$, $h_2(x)$ et de la permutation spécifiant l'entrelaceur qui associe la séquence permutée $\underline{a}^*$ à la séquence $\underline{a}$, spécifie un codeur que nous appelerons "turbocodeur". L'ensemble des séquences que peut produire un turbocodeur spécifié sera appelé "turbocode".

Dans la suite de la description, on appelle "premier codeur", le codeur convolutif récursif élémentaire qui produit la séquence $\underline{b}$ et "deuxième codeur", celui qui produit la séquence $\underline{c}$.

Les divisions polynomiales mises en oeuvre sont du type division suivant les puissances croissantes, bien connue de l'homme du métier. Elles utilisent l'arithmétique modulo 2. Les séquences $\underline{a}$, $\underline{b}$ et $\underline{c}$ sont des séquences binaires et dans le cas général les divisions qui définissent $\underline{b}$ et $\underline{c}$ présentent un reste.

Ce type de méthode de codage présente l'avantage de se prêter à un décodage itératif performant, peu complexe et peu coûteux.

Pour le mettre en oeuvre, plusieurs questions se posent :

I/ Comment choisir les polynômes $g(x)$, $h_1(x)$ et $h_2(x)$?

II/ Comment choisir la permutation des termes de la séquence $\underline{a}$ qui produit la séquence $\underline{a}^*$? Parmi les choix proposés, trois exemples d'entrelaceurs, c'est-à-dire d'opérateurs qui permutent les termes de la séquence $\underline{a}$, pour former la séquence $\underline{a}^*$, sont donnés ci-dessous :

A) dans le premier exemple, après avoir rangé les termes de $\underline{a}$ dans un tableau rectangulaire, successivement ligne par ligne et, pour chaque ligne, de gauche à droite, on forme la séquence $\underline{a}^*$ en prélevant dans ce tableau successivement les termes colonne après colonne et, pour chaque colonne, de haut en bas. Par exemple, dans le cas de séquences de six termes et d'utilisation d'un tableau de deux lignes de trois colonnes, l'entrelaceur transforme la séquence $\underline{a} = (a_0, a_1, a_2, a_3, a_4, a_5)$ en la séquence $\underline{a}^* = (a_0, a_3, a_1, a_4, a_2, a_5)$.

B) dans un deuxième exemple, le i-ième terme ($i = 0, 1, 2, ...$) a^*_i de la séquence $\underline{a}^*$ est choisi comme étant le terme a_j de la séquence $\underline{a}$, avec $j=s.i + t$ calculé modulo le nombre de termes de la séquence $\underline{a}$, s et t étant des entiers. Par exemple, si le nombre de termes de la séquence $\underline{a}$ est six et si $s = 5$ et $t = 3$, l'entrelaceur transforme la séquence $\underline{a} = (a_0, a_1, a_2, a_3, a_4, a_5)$ en la séquence $\underline{a}^* = (a_3, a_2, a_1, a_0, a_5, a_4)$.

C) dans le troisième exemple, la permutation choisie est aléatoire.

III/ Comment éviter que la division définissant $b(x)$ ne présente un reste ? et

IV/ Comment éviter que la division définissant $c(x)$ ne présente un reste ?

Répondre à ces deux dernières questions revient à résoudre un problème fréquemment mentionné dans la littérature sur les turbocodes qui est celui du "retour à l'état zéro" des codeurs convolutifs élémentaires définissant $\underline{b}$ et $\underline{c}$. Puisque les turbocodeurs possèdent deux codeurs récursifs élémentaires dont le deuxième utilise une permutation $\underline{a}^*$ de la séquence $\underline{a}$, on désire garantir que les polynômes $a(x)$ et $a^*(x)$ représentant la séquence d'information $u(x)$ soient simultanément divisibles par $g(x)$. Assurer cette condition de divisibilité de $a(x)$ est simple car il suffit de construire $a(x)$ à partir de $u(x)$ en complétant $u(x)$ par des symboles de bourrage en nombre égal au degré de $g(x)$ et dont la seule

fonction est de garantir l'absence de reste dans la division servant à produire *b(x)* à partir de *a(x)*.

Choisir une permutation produisant *a'(x)* à partir de *a(x)* qui garantisse à la fois la divisibilité de *a'(x)* par *g(x)* et de bonnes performances de correction d'erreur pour le turbocode ainsi spécifié est, en revanche, plus difficile.

Ce problème peut entraîner des disparités entre les probabilités d'erreurs après décodage des différents bits constituant *u(x)*.

Dans un article paru dans le volume 31, No. 1 de la revue "Electronics Letters" le 5 janvier 1995, MM. BARBULESCU et PIETROBON exposent qu'un entrelaceur peut être décrit en rangeant successivement et cycliquement les termes de la séquence $\underline{a}$ dans un nombre de séquences égal au degré du polynôme *g(x)* incrémenté de un, et que, dans ce cas, des permutations internes à chacune des séquences ainsi formées entraînent une égalité entre le reste de la division définissant la séquence $\underline{b}$ et celui de la division définissant la séquence $\underline{c}$.

Toutefois, et contrairement à ce qui est dit dans cet article, cette affirmation n'est vraie que si le polynôme *g(x)* est de la forme $\Sigma_{i=0 \text{ à } m} x^i$.

Dans un article intitulé "*Turbo-block-codes*", MM. C. BERROU, S. EVANO et G. BATTAIL exposent que, en rangeant les termes de la séquence $\underline{u}$, cycliquement, dans un nombre de colonnes égal à un multiple du degré *N0* du polynôme de type $x^n - 1$ de plus faible degré strictement positif qui soit divisible par *g(x)*, des permutations internes à chacune des colonnes ainsi formées entraînent que la somme du reste de la division définissant la séquence $\underline{b}$ et de celui de la division définissant la séquence $\underline{c}$ est nulle, si bien que la concaténation des séquences est divisible par $\underline{g}$.

Ce document, tout comme le précédent, restreint donc le choix des entrelaceurs à des formes particulières travaillant indépendamment sur des sous-ensembles des termes de la séquence $\underline{a}$ en leur appliquant des permutations internes. Il ne garantit toutefois pas que individuellement *a(x)* et *a'(x)* soient divisibles par *g(x)*. Seule est garantie la divisibilité par *g(x)* du

polynôme représentant la concaténation ($\underline{a}$, $\underline{a}^*$), consistant à mettre bout à bout les deux séquences $\underline{a}$ et $\underline{a}^*$.

Il s'ensuit une possible perte d'efficacité du décodeur puisque celui-ci n'est pas informé de l'état qu'avait le codeur à l'instant marquant la fin du calcul de $\underline{b}$ et le début du calcul de $\underline{c}$.

Aucun des articles cités ne proposent un choix effectif d'entrelaceur.

La présente invention entend remédier à ces inconvénients en proposant, d'une part, des familles d'entrelaceurs qui garantissent le retour à zéro à la fin de la séquence $\underline{c}$, lorsque la séquence $\underline{b}$ retourne à zéro et, d'autre part, en proposant un choix d'entrelaceurs plus large que celui proposé par les articles cités ci-dessus.

A cet effet, selon un premier aspect, la présente invention vise un procédé de permutation fournissant, à partir d'une séquence $\underline{a}$ de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$ divisible par un polynôme diviseur $g(x)$ et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^*$ associée à un polynôme $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^*$, ledit polynôme $a^*(x)$ étant destiné à être divisé par le polynôme $g(x)$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre entier M quelconque par l'entier $N0$, $N0$ étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par le polynôme $g(x)$,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de $N0$ colonnes et de M lignes, il comporte :

- au moins une permutation dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant

uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, et

- aucune permutation en dehors dudit ensemble.

Ci-dessus ont été introduites, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de $N0$ colonnes et de M lignes, les successions de permutations prises dans l'ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données.

Les inventeurs ont découvert que toutes ces successions de permutations et seulement celles-là, garantissent que pour tout polynôme $a(x)$ dont la division par $g(x)$ laisse un reste nul, le polynôme permuté $a^*(x)$ ait la même propriété.

L'ensemble des choix décrits dans la présente invention comporte les entrelaceurs exposés dans les deux articles mentionnés ci-dessus. Ainsi les performances exprimées en termes de taux d'erreur en fonction du rapport signal/bruit peuvent être améliorées sans augmenter la complexité du turbocodeur ni celle du turbodécodeur.

Selon un deuxième aspect, la présente invention vise un procédé de permutation fournissant, à partir d'une séquence $\underline{a}$ de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$ divisible par un polynôme $g(x)$ et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^{**}$ associée à un polynôme $a^{**}(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^{**}$, ledit polynôme $a^{**}(x)$ étant destiné à être divisé par un polynôme diviseur $g_2(x)$, pour former une séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre

entier *M* quelconque par l'entier *N0*, *N0* étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par le polynôme *g(x)*,

caractérisé en ce que, dans une représentation où les données binaires de la séquence *a* sont classées dans un tableau de *N0* colonnes et de *M* lignes, il comporte :

- au moins une permutation dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur *N0* et de polynôme générateur *g(x)*, permutant entre elles au moins deux des *N0* colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, produisant ainsi une séquence *a*(x)* et

- une permutation des colonnes dudit tableau qui transforme le polynôme *a*(x)* en le polynôme *a**(x)* divisible par le polynôme $g_2(x)$.

Ces successions de permutations garantissent que pour tout polynôme *a(x)* dont la division par *g(x)* laisse un reste nul, la division du polynôme permuté *a*(x)* par le polynôme $g_2(x)$ laisse aussi un reste nul.

Ce deuxième aspect de l'invention présente les mêmes avantages que le premier aspect.

Selon des caractéristiques particulières, la présente invention vise un procédé de codage caractérisé en ce qu'il comporte une opération de détermination de la séquence *a**, au cours de laquelle on met en oeuvre l'un des aspects du procédé de permutation de l'invention, tel que succinctement exposé ci-dessus.

Selon des caractéristiques particulières, la présente invention vise un procédé de codage tel que succinctement exposé ci-dessus, travaillant sur des données binaires u_l représentatives d'information et prenant en compte chaque polynôme diviseur, un premier polynôme multiplicatif $h_1(x)$ et un deuxième polynôme multiplicatif $h_2(x)$, qui comporte :

- une opération de constitution d'une "première" séquence, dite "séquence *a*" correspondant à un "premier" polynôme *a(x)* dont les coefficients

d'ordre croissant sont les données binaires de la première séquence $\underline{a}$, au cours de laquelle on constitue la première séquence $\underline{a}$ avec un nombre de données binaires u_i égal au produit d'un nombre entier M quelconque par l'entier $N0$, moins le degré du polynôme $g(x)$, d'une part, et un nombre égal au degré du polynôme $g(x)$ de valeurs binaires additionnelles choisies de telle manière que le polynôme $g(x)$ divise le polynôme $a(x)$, d'autre part,

- une opération de constitution de deux séquences binaires $\underline{b}$ et $\underline{c}$, chacune d'entre elles étant, à elle seule, représentative de la première séquence $\underline{a}$,

- ladite opération de permutation, travaillant sur des données binaires de la première séquence $\underline{a}$ pour former une séquence dite "permutée" $\underline{a}^*$ correspondant à un polynôme dit "permuté" $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence permutée $\underline{a}^*$ et qui soit divisible par le polynôme $g(x)$,

- la "deuxième" desdites séquences, dite "séquence $\underline{b}$" correspondant à un "deuxième" polynôme $b(x)$ dont les coefficients d'ordre croissant sont les données binaires de la deuxième séquence $\underline{b}$, le deuxième polynôme $b(x)$ étant égal au produit du premier polynôme $a(x)$ par le premier polynôme multiplicatif $h_1(x)$, le tout divisé par le polynôme diviseur $g(x)$,

- la "troisième" desdites séquences, dite "séquence $\underline{c}$" correspondant à un "troisième" polynôme $c(x)$ dont les coefficients d'ordre croissant sont les données binaires de la troisième séquence $\underline{c}$, le troisième polynôme $c(x)$ étant égal au produit du polynôme permuté $a^*(x)$ par le deuxième polynôme multiplicatif $h_2(x)$, le tout divisé par un polynôme diviseur.

Selon des caractéristiques particulières, la présente invention vise un procédé de décodage qui met en oeuvre un procédé de permutation tel que succinctement exposé ci-dessus.

Selon des caractéristiques particulières, dans le procédé de l'invention tel que succinctement exposé ci-dessus :

- ledit procédé comporte au moins une opération de permutation ne travaillant que sur les données binaires d'une desdites colonnes, et/ou

- ledit automorphisme est suivi par une permutation des colonnes dudit tableau qui rend $a^*(x)$ divisible par un autre polynôme dit "générateur" $g_2(x)$.

Grâce à ces dernières dispositions, lorsque ledit automorphisme est suivi d'une permutation supplémentaire notée P et qui a la propriété de transformer tout polynôme $a(x)$ ou $a^*(x)$, multiples de $g(x)$, en un nouveau polynôme $a^{**}(x)$ qui est, lui, multiple d'un autre polynôme $g_2(x)$ prédéterminé, l'obtention de $c(x)$ impliquant alors non plus la division de $a^*(x)$ par $g(x)$ mais celle de $a^{**}(x)$ par $g_2(x)$.

On notera toutefois que ceci n'est possible que si le plus petit entier $N2$ tel que $g_2(x)$ divise $x^{N2} - 1$ est égal au plus petit entier $N0$ tel que $g(x)$ divise $x^{N0} - 1$. Cette dernière condition est seulement nécessaire. La condition nécessaire et suffisante est que les deux codes cycliques binaires de longueur $N0 = N2$ engendrés respectivement par $g(x)$ et $g_2(x)$ puissent être obtenus l'un à partir de l'autre par permutation (voir à ce propos la page 234 du livre de Mme F.J. MAC WILLIAMS et M. N.J.A. SLOANE "*The theory or error-correcting codes*" publiée par l'éditeur North-Holland en 1977 et dont la septième impression a eu lieu en 1992).

Selon des caractéristiques particulières, dans le procédé de l'invention tel que succinctement exposé ci-dessus, ladite permutation est effectuée de la manière suivante :

$a^*(x) = a(x^e)$, modulo $x^{M.N0}-1$,

où $M.N0$ est un nombre impair et e est une valeur entière égale à une puissance de 2 , modulo $M.N0$. Ceci implique que M soit un nombre impair et que $g(x)$ soit choisi pour que $N0$ soit un nombre impair.

On observe ici que l'on dit, alors, que e appartient au cycle de 2, modulo $M.N0$.

Grâce à ces dispositions, la plupart des colonnes du tableau peuvent être déplacées par permutation, d'une part, et, dans ce choix restreint,

la distance minimale du turbocode est plus facilement analysable et donc peut être optimisée, d'autre part.

L'invention vise aussi :

A/ un entrelaceur adapté à fournir, à partir d'une séquence $\underline{a}$ de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$ divisible par un polynôme diviseur $g(x)$, et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^*$ associée à un polynôme $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^*$, ledit polynôme $a^*(x)$ étant destiné à être divisé par le polynôme $g(x)$ pour former une séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre entier M quelconque par l'entier $N0$, $N0$ étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par $g(x)$,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de $N0$ colonnes et de M lignes, il est adapté à mettre en oeuvre :

- au moins une permutation dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, et

- aucune permutation en dehors dudit ensemble.

B/ un entrelaceur adapté à fournir, à partir d'une séquence $\underline{a}$ de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$ divisible par un polynôme diviseur $g(x)$, et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^{**}$ associée à un polynôme $a^{**}(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^{**}$, ledit polynôme $a^{**}(x)$ étant destiné à être divisé par un polynôme diviseur $g_2(x)$, pour former une

séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre entier *M* quelconque par l'entier *N0*, *N0* étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par *g(x)*,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de *N0* colonnes et de *M* lignes, il est adapté à mettre en oeuvre :

- au moins une permutation dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur *N0* et de polynôme générateur *g(x)*, permutant entre elles au moins deux des *N0* colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, et

- une permutation des colonnes dudit tableau qui rend $a^*(x)$ divisible par le polynôme $g_2(x)$.

L'invention vise aussi :

- un dispositif de traitement de signaux représentatifs de parole, qui comporte un dispositif tel que succinctement exposé ci-dessus,

- un dispositif de transmission de données comportant un émetteur adapté à mettre en oeuvre un protocole de transmission par paquets, qui comporte un dispositif tel que succinctement exposé ci-dessus,

- un dispositif de transmission de données comportant un émetteur adapté à mettre en oeuvre le protocole de transmission par paquets ATM (mode de transfert asynchrone, "Asynchronous Transfer Mode"), qui comporte un dispositif tel que succinctement exposé ci-dessus,

- une station de réseau, qui comporte un dispositif tel que succinctement exposé ci-dessus,

- un dispositif de transmission de données comportant un émetteur émettant sur un canal non filaire, qui comporte un dispositif tel que succinctement exposé ci-dessus, et

- un dispositif de traitement de séquences de signaux représentatifs d'au plus mille données binaires, qui comporte un dispositif tel que succinctement exposé ci-dessus.

Ces dispositifs présentant les mêmes avantages que les procédés correspondants, ces avantages ne sont pas rappelés ici.

L'invention sera mieux comprise à la lecture de la description qui va suivre, faite en regard des dessins annexés dans lesquels :

- la figure 1 représente, schématiquement, un dispositif électronique incorporé dans un codeur selon la présente invention ;

- la figure 2 représente schématiquement un organigramme de fonctionnement d'un codeur tel qu'illustré en figure 1 ; et

- la figure 3 représente schématiquement un organigramme décrivant les étapes de détermination d'un entrelaceur mis en oeuvre dans le dispositif illustré en figure 1.

Dans la description qui va suivre, on appelle "données" aussi bien des symboles représentatifs d'information que des symboles additionnels ou redondants.

Avant d'entamer la description d'un mode particulier de réalisation, les fondements mathématiques de sa mise en oeuvre sont donnés ci-dessous.

Dans l'invention, $g(x)$, $h_1(x)$ et $h_2(x)$ étant prédéterminés, on rappelle que l'on souhaite que les séquences $\underline{b}$ et $\underline{c}$, définies respectivement par les divisions $b(x) = a(x).h_1(x)/g(x)$ et $c(x) = a^*(x).h_2(x)/g(x)$, ne présentent aucun reste.

A cet effet, $g(x) = 1 + \Sigma_{i=1 \text{ à } m-1} g_i.x^i + x^m$ étant un polynôme de degré m prédéterminé, on recherche le plus petit nombre $N0$ tel que $g(x)$ divise le polynôme $x^{N0} - 1$. On sait que ce nombre existe. Par exemple pour $g(x) = 1 + x + x^3$, $N0 = 7$.

Puis, en choisissant un nombre M quelconque, on choisit une longueur de séquence $\underline{a}$ égale à $M.N0$, ce qui revient à déterminer la longueur

(c'est-à-dire le nombre de données binaires) de la séquence $\underline{u}$ incorporée à la séquence $\underline{a}$ comme étant égal à *M.N0* moins le degré de *g(x)*.

Ainsi, pour former la séquence $\underline{a}$, on juxtapose à la séquence $\underline{u}$ formée de *k* données binaires u_l à transmettre, un nombre de données binaires additionnels égal au degré du polynôme *g(x)*, les données ajoutés garantissant l'absence de reste dans la division de *a(x)* par *g(x)*.

On rappelle que la division effectuée ici se fait, modulo 2, sur les coefficients des puissances croissantes de *a(x)*.

A titre d'exemple, si

la séquence $\underline{u}$ est la séquence *(1, 0, 0, 1, 0, 0)*, et

la séquence $\underline{g}$ est la séquence *(1, 1, 0, 1)*, la division s'écrit :

```
1 0 0 1 0 0 | 1 1 0 1
1 1 0 1     | 1 1 1 1 0 1
  1 1 0 1   |
    1 1 0 1 |
      1 1 0 1
        0 0 0 0
          1 1 0 1
-----------------
      0 0 0 0 0 0 0 0 1
```

ce qui s'écrit aussi : *(1, 0, 0, 1, 0, 0, 0, 0, 0) = (1, 1, 0, 1) x (1, 1, 1, 1, 0, 1) + (0, 0, 0, 0, 0, 0, 0, 0, 1)*,

soit encore *(1, 0, 0, 1, 0, 0, 0, 0, 1) = (1, 1, 0, 1) x (1, 1, 1, 1, 0, 1)*, en ajoutant, terme à terme, la séquence de reste *(0, 0, 0, 0, 0, 0, 0, 0, 1)* à la séquence $\underline{u}$ d'abord complétée par *m "0"*.

Ainsi, en substituant à la séquence $\underline{u}$ = *(1, 0, 0, 1, 0, 0)*, la séquence $\underline{a}$ = *(1, 0, 0, 1, 0, 0, 0, 0, 1)*, formée par cette addition, et dont les premières données binaires sont toutes les données binaires de la séquence $\underline{u}$, on garantit la divisibilité du polynôme *a(x)* par le polynôme *g(x)* associé à la séquence $\underline{g}$ = *(1, 1, 0, 1)*, et, par conséquent de $a(x).h_1(x)$ par *g(x)*, quelle que

soit la séquence $\underline{h_1}$ associée au polynôme $h_1(x)$, ce qui fournit la définition de la séquence $\underline{b}$ par $b(x) = a(x).h_1(x)/g(x)$.

Pour déterminer la séquence $a^*(x)$, qui possède, après permutation, les mêmes données binaires que la séquence $\underline{a}$, mais dans un ordre différent, on choisit un entrelaceur dont une représentation peut être donnée de la manière suivante : les données binaires de la séquence $\underline{a}$ étant classées dans un tableau de $N0$ colonnes et de M lignes, on effectue sur ces données au moins une permutation dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, et seulement une ou des permutations de cet ensemble.

En effet, les inventeurs ont découvert que seules les permutations qui peuvent être ainsi représentées garantissent que pour tout polynôme $a(x)$ dont la division par $g(x)$ ne présente aucun reste, le polynôme permuté $a^*(x)$ possède une division par $g(x)$ qui ne présente aucun reste.

Dans une telle succession, on peut, par exemple dans le cas où $N0 = 7$ et $g(x) = 1 + x + x^3$, trouver successivement :

- une permutation des données binaires de la première colonne,
- un permutation des données binaires de la troisième colonne,
- le remplacement de la deuxième colonne par la quatrième, le remplacement de la quatrième colonne par la deuxième, le remplacement de la cinquième colonne par la sixième et le remplacement de la sixième colonne par la cinquième.

En ce qui concerne l'automorphisme, en appelant Cg le code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, c'est-à-dire l'ensemble des multiples de $g(x)$, modulo $x^{N0}-1$, on considère les permutations des coordonnées de ce code qui transforment tout mot de ce code en un autre mot de ce code. L'ensemble des permutations de coordonnées qui possèdent

cette propriété a une structure de groupe et est appelé le groupe d'automorphisme de *Cg*.

Pour plus de détails, le lecteur pourra se référer à l'ouvrage de Madame F.J. MAC WILLIAMS et Monsieur N. J. A. SLOANE, "*The theory of error-correcting codes*" publié par l'éditeur North-Holland en 1977.

Parmi toutes ces permutations, les inventeurs ont sélectionné les permutations suivantes, qui présentent l'avantage de ne constituer qu'une petite famille dont tous les membres peuvent être testés pour choisir la permutation la plus performante.

Comme mentionné supra, on choisit *M* impair et *g(x)* tel que le nombre *N0* correspondant soit aussi impair. En écrivant les puissances successives de 2 modulo *M.N0*, on obtient ce qui s'appelle le cycle de 2, modulo *M.N0*. Dans ce cycle, en choisissant n'importe quel terme *e*, on effectue la permutation suivante :

le polynôme *a(x)* donnant, après permutation le polynôme $a^*(x)$, celui-ci est défini par $a^*(x) = a(x^e)$, ainsi, si $\underline{a}=(a_0, a_1, a_2, ..., a_{M.N0-1})$, la première donnée binaire de a^* est a_0, la seconde a_f, la troisième a_{2f}, ..., *f* étant l'inverse de *e*, modulo *M.N0*, et les multiples de *f* étant, eux-mêmes calculés modulo *M.N0*.

Par exemple en reprenant $g(x) = 1 + x + x^3$, et donc *N0* = 7, et en choisissant *M* = 5, on a *M. N0* = 35. Le cycle de 2 s'écrit alors :

[1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9, 18].

En prenant, par exemple $e = 2^8 = 11$, *f* vaudra 16 et la séquence $\underline{a}^*$ commence par les données binaires : $a_0, a_{16}, a_{32}, a_{13}, a_{29}, a_{10}$, etc ...

Ces permutations décrites ci-dessus et représentables par $a^*(x) = a(x^e)$ où *e* est dans le cycle de 2 modulo *M.N0*, $a(x^e)$ étant pris modulo $x^{M.N0} - 1$, forment une petite famille dont tous les membres peuvent être testés pour choisir le plus performant.

La logique de sélection est la suivante : on choisit d'abord *g(x)* de degré *m*. Ce choix fixe la valeur de *N0* comme étant le plus petit entier tel que *g(x)* divise $x^{N0}-1$. On choisit alors $h_1(x)$ et $h_2(x)$ de degré quelconque mais

préférentiellement au plus égal au degré m de $g(x)$ car le maximum des degrés de ces trois polynômes $g(x)$, $h_1(x)$ et $h_2(x)$ est un élément déterminant dans la complexité du décodeur. On choisit ensuite un entier impair M et on calcule le cycle de deux modulo $M.N0$. On choisit alors un élément e dans ce cycle de 2 pour spécifier $a^*(x) = a(x^e)$ modulo $X^{M.N0-1}-1$ à partir de $a(x)$ et on effectue diverses opérations de test sur le turbocode associé à l'entrelaceur ainsi défini.

Prenons par exemple $g(x) = 1 + x + x^3$, ceci impose $N0 = 7$. Choisissons aussi $h_1(x) = 1 + x + x^2 + x^3$, $h_2(x) = 1 + x^2 + x^3$ et $M = 21$ ceci entraîne $M.N0 = 147$ et permet de calculer le cycle de deux modulo 147 comme étant {1, 2, 4, 8, 32, 64, 128, 109, 71, 142, 137, 127, 107, 67, 134, 121, 95, 43, 86, 25, 50, 100, 53, 106, 65, 130, 113, 79, 11, 22, 44, 88, 29, 58, 116, 85, 23, 46, 92, 37, 74}.

En testant successivement les polynômes $a(x)$ divisibles par $g(x)$ de poids égal à 2, 3, 4 et 5, on conclut que le choix $e = 25$ est "prometteur" car les $a(x)$ de poids 2 correspondent alors à une séquence codée $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ de poids ≥ 26, les $a(x)$ de poids 3 correspondent alors à une séquence codée $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ de poids ≥ 24, les $a(x)$ de poids 4 correspondent alors à une séquence codée $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ de poids ≥ 26, les $a(x)$ de poids 5 correspondent alors à une séquence codée $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ de poids ≥ 30 ...

Cela semble indiquer une distance minimale égale à 24, ce qui est la meilleure valeur que l'on puisse obtenir suivant la méthode exposée ci-dessus pour $N0 = 7$ et $M = 21$ avec $g(x)$, $h_1(x)$ et $h_2(x)$ comme indiqué ci-dessus.

Un autre choix possible est $g(x) = 1 + x + x^4$ qui impose $N0 = 15$. Choisissons aussi $h_1(x) = 1 + x + x^2 + x^4$, $h_2(x) = 1 + x^3 + x^4$ et $M = 27$. Ceci entraîne $M.N0 = 405$ et permet de calculer le cycle de 2 modulo 405 comme étant {1, 2, 4, 8, 16, ..., 304, 203}. Il comprend 108 nombres.

En travaillant par élimination successive, on conclut que les choix $e = 151$, $e = 362$ et $e = 233$ sont particulièrement prometteurs.

En particulier pour $e = 151$, les inventeurs ont testé les polynômes $a(x)$ divisibles par $g(x)$ et de poids égal successivement à 2, 3, 4, 5, 6 et 7.

Lorsque le poids de $a(x)$ est $W(a(x))$, le poids minimal de $W(\underline{y})$ correspondant est indiqué dans la table :

$W(a(x))$	$W(\underline{y}) \geq$
2	54
3	42
4	44
5	48
6	54
7	54

De la même manière, pour $e = 362$, lorsque le poids de $a(x)$ est $W(a(x))$, le poids minimal de $W(\underline{y})$ correspondant est indiqué dans la table :

$W(a(x))$	$W(\underline{y}) \geq$
2	54
3	42
4	42
5	50
6	56
7	54

Selon l'invention, les nombres e, non congrus à 1, modulo $N0$, peuvent être particulièrement utilisés.

La description d'un mode particulier de réalisation de la présente invention va maintenant se poursuivre en regard des figures 1 à 3.

La figure 1 illustre schématiquement la constitution d'une station de réseau ou station de codage informatique 101, sous forme de schéma

synoptique. Cette station comporte un clavier 111, un écran 109, une source d'informations externe 110, un émetteur hertzien 106, conjointement reliés à un port d'entrée/sortie 103 d'une carte de traitement 101.

La carte de traitement 101 comporte, reliés entre eux par un bus d'adresses et de données 102 :

- une unité centrale de traitement 100 ;
- une mémoire vive RAM 104 ;
- une mémoire morte ROM 105 ;
- le port d'entrée/sortie 103 ;

Chacun des éléments illustrés en figure 1 est bien connu de l'homme du métier des micro-ordinateurs et des systèmes de transmission et, plus généralement, des systèmes de traitement de l'information. Ces éléments communs ne sont donc pas décrits ici. On observe, cependant, que :

- la source d'informations 110 est, par exemple, un périphérique d'interface, un capteur, un démodulateur, une mémoire externe ou un autre système de traitement d'information (non représenté), et est préférentiellement adaptée à fournir des séquences de signaux représentatifs de parole, sous forme de séquences d'au plus mille données binaires,

- l'émetteur hertzien 106 est adapté à mettre en oeuvre un protocole de transmission par paquets, et plus particulièrement le protocole ATM, Asynchronous Transfer Mode, sur un canal non filaire.

On observe, en outre, que le mot "registre" utilisé dans la description désigne, dans chacune des mémoires, aussi bien une zone mémoire de faible capacité (quelques données binaires) qu'une zone mémoire de grande capacité (permettant de stocker un programme entier).

La mémoire vive 104 conserve des données, des variables et des résultats intermédiaires de traitement, dans des registres de mémoire portant, dans la description, les mêmes noms que les données dont ils conservent les valeurs. La mémoire vive 104 comporte notamment :

- un registre "*données_primaires*" dans lequel sont conservées, dans l'ordre de leur arrivée sur le bus 102, les données binaires en provenance de la source d'information 110,

- un registre "*données_permutées*" dans lequel sont conservées, dans l'ordre de leur arrivée sur le bus 102, les données binaires permutées, comme décrit en regard de la figure 2,

- un registre "*nb_données*" qui conserve un nombre entier correspondant au nombre de données binaires dans le registre "*données_binaires*",

- un registre "*reste_intermédiaire*" dans lequel sont conservés successivement les restes intermédiaires de la division,

- un registre "*reste_final*" dans lequel sont conservées des données binaires complémentaires, et

- un registre "*a, b, c*" dans lequel sont conservées, dans l'ordre de leur détermination par l'unité centrale 100, les données binaires des séquences secondaires.

La mémoire morte 105 est adaptée à conserver, dans des registres qui, par commodité, possèdent les mêmes noms que les données qu'ils conservent :

- le programme de fonctionnement de l'unité centrale de traitement 100, dans un registre "*program*",

- la séquence g, dans un registre "*g*",

- le degré de $g(x)$, dans un registre "*degré*"

- la séquence h_1, dans un registre "h_1",

- la séquence h_2, dans un registre "h_2",

- la valeur de *N0*, dans un registre "*N0*",

- la valeur de *M*, dans un registre "*M*", et

- le tableau définissant l'entrelaceur, dans un registre "*entrelaceur*".

L'unité centrale de traitement 100 est adaptée à mettre en oeuvre l'organigramme décrit en figure 2.

En figure 2, qui représente le fonctionnement d'un codeur tel qu'illustré en figure 1, on observe qu'après une opération d'initialisation 300, au cours de laquelle, les registres de la mémoire vive 104 sont initialisés (*nb_données* = "0"), au cours d'une opération 301, l'unité centrale 100 attend de recevoir, puis reçoit une donnée binaire à transmettre, la positionne en mémoire vive 104, dans le registre "*données_primaires*" et incrémente le compteur "*nb_données*".

Ensuite, au cours d'un test 302, l'unité centrale 100 détermine si le nombre entier conservé dans le registre "*nb_données*" est égal, ou non au produit *M.N0* auquel est soustrait le degré de *g(x)*, *M*, *N0* et le degré *m* de *g(x)* étant des valeurs conservées en mémoire morte 105.

Lorsque le résultat du test 302 est négatif, l'opération 301 est réitérée. Lorsque le résultat du test 302 est positif, au cours d'une l'opération 303, la division du polynôme *u(x)* associé à la séquence de données binaires conservée dans le registre "*données_primaires*" par le polynôme *g(x)* est effectuée, jusqu'au dernier terme (de plus haut degré) de *u(x)*, en mettant en oeuvre, à cet effet, le registre "*reste_intermédiaire*", le reste de cette division est mis en mémoire dans le registre "*reste_final*". Le résultat de cette division est mis en mémoire dans le registre "*a*, *b*, *c*", et correspond aux premières données binaires de la séquence *b*.

Ensuite, au cours d'une opération 304, les données binaires conservées dans le registre "*reste_final*" sont juxtaposées à la fin de la séquence *u* pour former la séquence *a*, à l'exception de la donnée binaire qui correspond au plus bas degré de polynôme, cette donnée binaire valant d'ailleurs nécessairement "0". Les données binaires de la séquence *a* sont mises en mémoire dans le registre "*a*, *b*, *c*".

Ensuite, au cours d'une opération 305, la division effectuée au cours de l'opération 303 est achevée, avec les données additionnelles ajoutées au cours de l'opération 304 et la séquence *b* est complétée dans le registre "*a*, *b*, *c*".

Puis, au cours d'une opération 306, les données binaires de la séquence a sont successivement lues dans le registre "$\underline{a}$, $\underline{b}$, $\underline{c}$", dans l'ordre décrit par le tableau "*entrelaceur*" conservé en mémoire morte 105. Les données qui résultent successivement de cette lecture sont mises en mémoire dans le registre "*données_permutées*" de la mémoire vive 104.

Ensuite, au cours d'une opération 307, la division du polynôme $a^*(x)$ associé à la séquence de données binaires permutées conservée dans le registre "*données_permutées*" par le polynôme $g(x)$ est effectuée, en mettant en oeuvre, à cet effet, le registre "*reste_intermédiaire*". Le résultat de cette division est mis en mémoire dans le registre "$\underline{a}$, $\underline{b}$, $\underline{c}$", et correspond aux données binaires de la séquence $\underline{c}$.

Au cours d'une opération 308, les séquences $\underline{b}$ et $\underline{c}$ sont déterminées en effectuant le produit des polynômes associés aux séquences $\underline{b}$ et $\underline{c}$ conservées dans le registre "$\underline{a}$, $\underline{b}$, $\underline{c}$" de la mémoire vive 104, respectivement par les polynômes $h_1(x)$ et $h_2(x)$.

On observe que, grâce à l'invention, on gagne des éléments de mémoire en effectuant la division par $g(x)$ avant la multiplication par $h_1(x)$ ou $h_2(x)$.

Au cours d'une opération 309, les séquences $\underline{a}$, $\underline{b}$, et $\underline{c}$ sont émises en utilisant, à cet effet, l'émetteur 106. Ensuite, les registres de la mémoire 104 sont à nouveau initialisés, en particulier le compteur *Nb_données* est remis à "0" et l'opération 301 est réitérée.

On observe ici qu'en variante, au cours de l'opération 309, la séquence $\underline{a}$ est émise intégralement, mais seulement un sous-ensemble, par exemple une donnée sur deux, de chacune des séquences $\underline{b}$ et $\underline{c}$ est émis. Cette variante est connue de l'homme du métier sous le nom de poinçonnage.

En ce qui concerne le décodage, on observe qu'en connaissant les polynômes $g(x)$, $h_1(x)$, $h_2(x)$ et l'entrelaceur qui, à partir de la séquence $\underline{a}$ fournissent la séquence permutée $\underline{a}^*$, l'homme du métier sait, sans problème technique, réaliser le décodeur adapté au décodage et à la correction d'erreur

affectant le triplet de séquences ($\underline{a}$, $\underline{b}$, $\underline{c}$) en utilisant l'entrelaceur considéré ci-dessus et, éventuellement le désentrelaceur correspondant.

A cet effet, il peut se référer :

- à l'article de MM. L.R. BAHL, J. COCKE, F. JELINEK et J. RAVIV intitulé "*Optimal decoding of linear codes for minimizing symbol error rate*", publié dans la revue IEEE Transactions on Information Theory, en mars 1974 ;

- à l'article de MM. J. HAGENAUER, E. OFFER et L. PAPKE intitulé "*Iterative decoding of binary block and convolutional codes*" publié dans la revue IEEE Transactions on Information Theory, en mars 1996 ;

- à l'article de MM. J. HAGENAUER et P. HOEHER intitulé "*A viterbi algorithm with soft decision outputs and its applications*", publiée avec les compte-rendus de la conférence IEEE GLOBECOM, pages 1680-1686, en novembre 1989 ;

- à l'article de MM. J. HAGENAUER, P. ROBERTSON et L. PAPKE intitulé "*Iterative (turbo)decoding of systematic convolutional codes with the MAP and SOVA algorithms*", publiée par la revue Informationstechnische Gesellschaft (ITG) Fachbericht, pages 21 - 29, octobre 1994 ; et

- à l'article de MM. C. BERROU, S. EVANO et G. BATTAIL, intitulé "*Turbo-block-codes*" et publié avec les compte-rendus du séminaire "turbo coding" organisé par l'Institut de Technologie de Lund (Suède) (Département d'électronique appliquée) en août 1996.

En figure 3, sont représentées les étapes de détermination de la valeur de e spécifiant l'entrelaceur à utiliser. Ces étapes peuvent être réalisées par le dispositif de codage illustré en figure 1, les registres "*N0*", "*M*" et "*entrelaceur*" se trouvant, dans ce cas, en mémoire vive 104 et non en mémoire morte 105 et quatre registres "*d*", "d_{max}", "*e*" et "*j*" étant ajoutés en mémoire vive 104.

Au cours d'une opération 501, $g(x) = 1 + \Sigma_{i=1 \text{ à } m-1} g_i x^i + x^m$ étant le polynôme de degré m prédéterminé qui correspond à la séquence $\underline{g}$, on

recherche le plus petit nombre entier strictement positif *N0* tel que *g(x)* divise le polynôme $x^{N0} - 1$. On sait que ce nombre existe. Par exemple pour $g(x) = 1 + x + x^3$, *N0* = 7. A cet effet, on effectue successivement les divisions des polynômes $x^j - 1$ par g(x) en commençant par une valeur de *i* égale au degré *m* de g(x) et en incrémentant progressivement *i*, avec un pas d'incrémentation de 1, jusqu'à ce que le reste de la division soit nul, modulo 2. Lorsque le reste est nul, la valeur de *i* est mise dans le registre *N0*. On rappelle que la division effectuée ici se fait, modulo 2, sur les coefficients des puissances croissantes de $x^j - 1$.

Puis, en choisissant un nombre *M* impair, de telle manière que le produit *M.N0* soit supérieur ou égal au nombre de données binaires u_l qui doivent être transmises dans la même trame, additionné au degré *m* de *g(x)*, au cours d'une opération 502, on choisit une longueur de séquence $\underline{a}$ égale à *M.N0*. ce qui revient à déterminer la longueur (c'est-à-dire le nombre de données binaires) de la séquence $\underline{u}$ incorporée à la séquence $\underline{a}$ comme étant égal à *M.N0* moins le degré *m* de *g(x)*.

Puis, au cours des opérations 503 à 509, l'unité centrale 100 détermine si l'entrelaceur associé à *e* est à prendre en compte, ce qui veut dire qu'il n'y a pas de séquence $\underline{a}$ de poids faible pour laquelle la séquence $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ possède, elle aussi, un poids faible.

Dans le mode de réalisation décrit et représenté, la détermination de $\underline{a}^*$ consistera à remplacer $\underline{a} = (a_0, a_1, \ldots)$ par $\underline{a}^* = (a_0, a_f, a_{2f}, \ldots)$ où les multiples de *f* sont calculés modulo *M.N0*. Lorsque *f* est égal à une puissance de 2 différente de 1, modulo *M.N0*, cette permutation est bien du type annoncé. Elle peut, en effet, être représentée par une permutation qui ne permute des données binaires qu'à l'intérieur de chaque colonne du tableau, suivie, lorsque *f* est égal à une puissance de 2 différente de 1, modulo *N0*, d'une permutation d'au moins deux des colonnes entre elles, cette dernière permutation étant un automorphisme du code cyclique binaire de longueur *N0* et de polynôme générateur *g(x)*. Lorsque *f* est égal à 1, modulo *N0*, cette permutation portant

sur les colonnes est la permutation "triviale" ou permutation "identité", c'est-à-dire celle qui maintient la position des colonnes dans le tableau.

A cet effet, dans ce mode particulier de réalisation de la présente invention, au cours de l'opération 503, l'unité centrale 100 détermine les puissances successives de 2 modulo *M.N0*, pour obtenir ce qui s'appelle le cycle de 2, modulo *M.N0*, ce cycle étant achevé dès qu'une des puissances de 2 est égale à 1, modulo *M.N0*. Le nombre de termes *j* de ce cycle est mémorisé dans le registre "*j*".

Puis, au cours de l'opération 504, on initialise les valeurs intermédiaires *l* et d_{max} conservées dans les registres "*l*" et "d_{max}", respectivement à la valeur "1" et à la valeur "0".

Ensuite, au cours d'une opération 505, on incrémente de "1" la valeur de *l* et on prend la *l*-ième valeur du cycle de 2, modulo *M.N0*.

Puis, au cours d'une opération 506, si cette valeur n'est pas égale à 1 modulo *M.N0*, on détermine le poids de la séquence $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ pour les séquences $\underline{a}$ de poids faible, avec la permutation définie par $a^*(x) = a(x^e)$ (ainsi, si $\underline{a}=(a_0, a_1, a_2, ..., a_{M.N0-1})$, la première donnée binaire de a^* est a_0, la seconde a_f, la troisième a_{2f}, ..., comme exposé supra, l'indice étant calculé modulo *M.N0*).

A cet effet, puisque la distance entre deux séquences est le poids (c'est-à-dire le nombre de données binaires non nulles) de la séquence constituée par différence des données binaires homologues de ces séquences, on se limite à l'analyse de la distance des séquences avec la séquence nulle, on énumère les polynômes *a(x)* par poids croissant, on mesure la somme des poids des séquences d'un même triplet $(\underline{a}, \underline{b}, \underline{c})$ et on cherche le poids minimal pour *e* donné, pour les séquences $\underline{a}$ de poids faible, et, une fois tous ces poids minimaux déterminés pour *e* dans le cycle de 2, la valeur de *e* qui correspond au poids le plus élevé.

La distance est ensuite mémorisée dans le registre "*d*" de la mémoire vive 104. Ensuite, si, au cours d'un test 507, la valeur conservée dans le registre "*d*" est supérieure à la valeur conservée dans le registre "d_{max}", alors

au cours de l'opération 508, la valeur du registre "d_{max}" est modifiée pour prendre la valeur d et la valeur du l-ième élément du cycle considéré est mise en mémoire dans le registre "e".

A la suite de l'opération 508 ou, lorsque le résultat du test 507 est négatif, tant que la valeur de l est inférieure à j, test 509, l'opération 505 est réitérée.

Le tableau "*entrelaceur*" est alors constitué de la manière suivante :

$a^*(x) = a(x^e)$, ainsi, si $\underline{a}=(a_0, a_1, a_2, ..., a_{M.N0-1})$, la première donnée binaire de a^* est a_0, la seconde a_1, la troisième a_{2l}, ..., comme exposé supra, l'indice étant calculé modulo $M.N0$.

Selon une variante non représentée, le triplet $(\underline{a}, \underline{b}, \underline{c})$ est construit de la manière suivante :

- $a(x)$ et $b(x) = a(x).h_1(x)/g(x)$ sont définis comme ci-dessus,

- étant donné un polynôme $g_2(x)$ choisi tel que le plus petit entier $N2$ tel que $g_2(x)$ divise le polynôme $x^{N2} - 1$ soit égal au plus petit entier $N0$ tel que $g(x)$ divise le polynôme $x^{N0} - 1$, on choisit une permutation P qui transforme tout mot du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$ en un mot du code cyclique binaire de longueur $N2$ et de polynôme générateur $g_2(x)$. On note qu'une telle permutation n'existe que pour les polynômes $g_2(x)$ engendrant des codes cycliques équivalent à Cg. Cette vérification est bien connue de l'homme du métier et nous renvoyons pour cela à la page 234 du livre de Mme F.J. MAC WILLIAMS et M. N.J.A. SLOANE mentionné ci-dessus ;

- selon l'invention, la permutation qui, à partir de la séquence $\underline{a}$, associée au polynôme $a(x)$ divisible par $g(x)$, produit la séquence $\underline{a}^{**}$, associée au polynôme $a^{**}(x)$, divisible par $g_2(x)$ est alors produite par une permutation quelconque produisant à partir de $a(x)$ une première séquence $a^*(x)$ divisible par $g(x)$ comme expliqué ci-dessus, suivie de la permutation P que nous venons d'introduire et qui agit sur les colonnes du tableau à M lignes et $N0$

colonnes contenant tout d'abord $\underline{a}$ et ensuite $\underline{a}^*$ pour permuter ces colonnes entre elles et produire $\underline{a}^{**}$

La portée de l'invention ne se limite pas aux modes de réalisation décrits et représentés mais s'étend, bien au contraire, aux modifications et perfectionnements à la portée de l'homme du métier.

En particulier, le passage à des rendements de un quart ou moins, par adjonction d'un ou de plusieurs entrelaceurs supplémentaires se fait pour chaque entrelaceur, par application des principes énoncés ci-dessus. Dans tous ces cas, l'usage du poinçonnage peut être fait pour élever le rendement du code.

De plus, la réalisation des dispositifs objets de la présente invention est avantageusement faite en mettant en oeuvre, pour effectuer les calculs arithmétiques, de multiplication polynomiale, de division polynomiale, la fonction d'entrelacement et les fonctions de décodage élémentaire, des circuits dédiés ne comportant pas de processeur (un tel processeur peut, néanmoins être utilisé pour le contrôle du fonctionnement de ces dispositifs). L'utilisation de tels circuits dédiés permet, en effet, d'atteindre des débits d'informations plus élevés.

REVENDICATIONS

1. Procédé de permutation fournissant, à partir d'une séquence $\underline{a}$ de données binaires, représentatives d'une grandeur physique, associée à un polynôme $a(x)$, divisible par un polynôme diviseur $g(x)$, et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^*$ associée à un polynôme $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^*$, ledit polynôme $a^*(x)$ étant destiné à être divisé par le polynôme $g(x)$, pour former une séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre entier M quelconque par l'entier $N0$, $N0$ étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par le polynôme $g(x)$,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de $N0$ colonnes et de M lignes, il comporte :

- au moins une permutation (306) dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, et
- aucune permutation en dehors dudit ensemble.

2. Procédé de permutation fournissant, à partir d'une séquence $\underline{a}$ de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$, divisible par un polynôme diviseur $g(x)$, et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^{**}$ associée à un polynôme $a^{**}(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^{**}$, ledit polynôme $a^{**}(x)$ étant destiné à être divisé par un polynôme diviseur $g_2(x)$, pour former une séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données

binaires égal au produit d'un nombre entier *M* quelconque par l'entier *N0*, *N0* étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par *g(x)*,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de *N0* colonnes et de *M* lignes, il comporte :

- au moins une permutation (306) dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur *N0* et de polynôme générateur *g(x)*, permutant entre elles au moins deux des *N0* colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, produisant ainsi une séquence *a*(x)* et

- une permutation des colonnes dudit tableau qui transforme le polynôme *a*(x)* en le polynôme *a**(x)* divisible par le polynôme $g_2(x)$.

3. Procédé de permutation selon l'une quelconque des revendications 1 ou 2, caractérisé en ce qu'il comporte au moins une opération de permutation (306) ne travaillant que sur les données binaires d'une desdites colonnes.

4. Procédé de permutation selon l'une quelconque des revendications 1 à 3, caractérisé en ce que ladite opération de permutation (306) est effectuée de la manière suivante :

$a^*(x) = a(x^e)$, modulo $x^{M.N0}-1$,

où *e* est une valeur entière égale à une puissance de 2, modulo *M.N0* et *M* est un nombre impair.

5. Procédé de permutation selon l'une quelconque des revendications 1 à 4, caractérisé en ce qu'il comporte un opération de détermination, à partir de la séquence de données binaires $\underline{a}$, d'au moins une seconde séquence permutée en mettant en oeuvre un procédé selon l'une quelconque des revendications 1 à 4.

6. Procédé de permutation selon l'une quelconque des revendications 1 à 5, caractérisé en ce qu'il comporte une opération de

transmission (309) d'une part de la séquence $\underline{a}$, et, d'autre part, d'un sous-ensemble des données des autres séquences.

7. Procédé de codage caractérisé en ce qu'il comporte une opération de détermination de la séquence $\underline{a}^*$, au cours de laquelle on met en oeuvre un procédé de permutation selon l'une quelconque des revendications 1 à 6.

8. Procédé de codage selon la revendication 7, travaillant sur des données binaires u_i représentatives d'information et prenant en compte chaque polynôme diviseur, un premier polynôme multiplicatif $h_1(x)$ et un deuxième polynôme multiplicatif $h_2(x)$, caractérisé en ce qu'il comporte :

- une opération de constitution d'une "première" séquence (304), dite "séquence $\underline{a}$" correspondant à un "premier" polynôme $a(x)$ dont les coefficients d'ordre croissant sont les données binaires de la première séquence $\underline{a}$, au cours de laquelle on constitue la première séquence $\underline{a}$ avec un nombre de données binaires u_i égal au produit d'un nombre entier M quelconque par l'entier $N0$, moins le degré du polynôme $g(x)$, d'une part, et un nombre égal au degré du polynôme $g(x)$ de valeurs binaires additionnelles choisies de telle manière que le polynôme $g(x)$ divise le polynôme $a(x)$, d'autre part,

- une opération de constitution (308) de deux séquences binaires $\underline{b}$ et $\underline{c}$, chacune d'entre elles étant, à elle seule, représentative de la première séquence $\underline{a}$,

- ladite opération de permutation (306), travaillant sur des données binaires de la première séquence $\underline{a}$ pour former une séquence dite "permutée" $\underline{a}^*$ correspondant à un polynôme dit "permuté" $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence permutée $\underline{a}^*$ et qui soit divisible par le polynôme diviseur $g(x)$,

- la "deuxième" desdites séquences, dite "séquence $\underline{b}$" correspondant à un "deuxième" polynôme $b(x)$ dont les coefficients d'ordre croissant sont les données binaires de la deuxième séquence $\underline{b}$, le deuxième

polynôme $b(x)$ étant égal au produit du premier polynôme $a(x)$ par le premier polynôme multiplicatif $h_1(x)$, le tout divisé par le polynôme diviseur $g(x)$,

- la "troisième" desdites séquences, dite "séquence $\underline{c}$" correspondant à un "troisième" polynôme $c(x)$ dont les coefficients d'ordre croissant sont les données binaires de la troisième séquence $\underline{c}$, le troisième polynôme $c(x)$ étant égal au produit du polynôme permuté $a^*(x)$ par le deuxième polynôme multiplicatif $h_2(x)$, le tout divisé par un polynôme diviseur.

9. Procédé de décodage, caractérisé en ce qu'il met en oeuvre un procédé de permutation selon l'une quelconque des revendications 1 à 6.

10. Entrelaceur (101) adapté à fournir, à partir d'une séquence $\underline{a}$ de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$ divisible par un polynôme diviseur $g(x)$, et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^*$ associée à un polynôme $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^*$, ledit polynôme $a^*(x)$ étant destiné à être divisé par le polynôme $g(x)$ pour former une séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre entier M quelconque par l'entier $N0$, $N0$ étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par $g(x)$,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de $N0$ colonnes et de M lignes, il est adapté à mettre en oeuvre :

- au moins une permutation (306) dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, et

- aucune permutation en dehors dudit ensemble.

11. Entrelaceur (101) adapté à fournir, à partir d'une séquence $\underline{a}$, de données binaires représentatives d'une grandeur physique, associée à un polynôme $a(x)$ divisible par un polynôme diviseur $g(x)$, et dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}$, une séquence permutée $\underline{a}^{**}$ associée à un polynôme $a^{**}(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence $\underline{a}^{**}$, ledit polynôme $a^{**}(x)$ étant destiné à être divisé par un polynôme diviseur $g_2(x)$, pour former une séquence de données binaires $\underline{c}$, et $\underline{a}$ possédant un nombre de données binaires égal au produit d'un nombre entier M quelconque par l'entier $N0$, $N0$ étant le plus petit entier tel que $x^{N0} - 1$ soit divisible par $g(x)$,

caractérisé en ce que, dans une représentation où les données binaires de la séquence $\underline{a}$ sont classées dans un tableau de $N0$ colonnes et de M lignes, il est adapté à mettre en oeuvre :

- au moins une permutation (306) dans un ensemble de permutations comportant, d'une part, les automorphismes du code cyclique binaire de longueur $N0$ et de polynôme générateur $g(x)$, permutant entre elles au moins deux des $N0$ colonnes du tableau et, d'autre part, les permutations travaillant uniquement sur des données d'une même colonne et permutant entre elles au moins deux desdites données, produisant ainsi une séquence $a^{*}(x)$ et

- une permutation des colonnes dudit tableau qui transforme le polynôme $a^{*}(x)$ en le polynôme $a^{**}(x)$ divisible par le polynôme $g_2(x)$.

12. Entrelaceur selon l'une quelconque des revendications 10 ou 11, caractérisé en ce que l'entrelaceur est adapté à ce que ladite permutation (306) comporte au moins une opération de permutation ne travaillant que sur les données binaires d'une desdites colonnes.

13. Entrelaceur selon l'une quelconque des revendications 10 à 12, caractérisé en ce que l'entrelaceur (101) est adapté à effectuer ladite permutation de la manière suivante :

$a^{*}(x) = a(x^{P})$, modulo $x^{M.N0}-1$,

où e est une valeur entière égale à une puissance de 2, modulo $M.N0$ et M est un nombre impair.

14. Entrelaceur selon l'une quelconque des revendications 10 à 13, caractérisé en ce qu'il comporte, en outre, au moins un deuxième entrelaceur selon l'une quelconque des revendications 10 ou 11, chaque nouvel entrelaceur étant adapté à fournir, à partir de la séquence de données binaires $\underline{a}$, une nouvelle séquence permutée.

15. Entrelaceur (101) selon l'une quelconque des revendications 10 à 14, caractérisé en ce qu'il comporte un moyen de transmission (106) adapté à transmettre, d'une part, la séquence $\underline{a}$, et, d'autre part, un sous-ensemble des données des autres séquences.

16. Dispositif de codage caractérisé en ce qu'il comporte un entrelaceur (101) selon l'une quelconque des revendications 10 à 15.

17. Dispositif de codage selon la revendication 16, travaillant sur des données binaires u_i et prenant en compte chaque polynôme diviseur $g(x)$, un premier polynôme multiplicatif $h_1(x)$ et un deuxième polynôme multiplicatif $h_2(x)$, caractérisé en ce qu'il comporte :

- un moyen de constitution d'une "première" séquence (101, 304), dite "séquence $\underline{a}$" correspondant à un "premier" polynôme $a(x)$ divisible par le polynôme $g(x)$, dont les coefficients d'ordre croissant sont les données binaires de la première séquence $\underline{a}$, adapté à constituer la première séquence $\underline{a}$ avec un nombre de données binaires u_i égal au produit d'un nombre entier M quelconque par l'entier $N0$, moins le degré du polynôme $g(x)$, d'une part, et un nombre égal au degré du polynôme $g(x)$ de valeurs binaires additionnelles choisies de telle manière que le polynôme $g(x)$ divise le polynôme $a(x)$, d'autre part,

- un moyen de constitution (101, 308) de deux séquences binaires $\underline{b}$ et $\underline{c}$, chacune d'entre elles étant, à elle seule, représentative de la première séquence $\underline{a}$,

- l'entrelaceur, adapté à effectuer une permutation des données binaires de la première séquence $\underline{a}$ pour former une séquence dite "permutée" $\underline{a}'$

correspondant à un polynôme dit "permuté" $a^*(x)$ dont les coefficients d'ordre croissant sont les données binaires de la séquence permutée $\underline{a}^*$ et qui est divisible par le polynôme $g(x)$,

et en ce que :

- la "deuxième" desdites séquences, dite "séquence $\underline{b}$" correspondant à un "deuxième" polynôme $b(x)$ dont les coefficients d'ordre croissant sont les données binaires de la deuxième séquence $\underline{b}$, le deuxième polynôme $b(x)$ étant égal au produit du premier polynôme $a(x)$ par le premier polynôme multiplicatif $h_1(x)$, le tout divisé par le polynôme diviseur $g(x)$,

- la "troisième" desdites séquences, dite "séquence $\underline{c}$" correspondant à un "troisième" polynôme $c(x)$ dont les coefficients d'ordre croissant sont les données binaires de la troisième séquence $\underline{c}$, le troisième polynôme $c(x)$ étant égal au produit du polynôme permuté $a^*(x)$ par le deuxième polynôme multiplicatif $h_2(x)$, le tout divisé par un polynôme diviseur.

18. Dispositif de décodage, caractérisé en ce qu'il met en oeuvre un entrelaceur selon l'une quelconque des revendications 10 à 15 et/ou un désentrelaceur correspondant à cet entrelaceur.

19. Dispositif de traitement de signaux représentatifs de parole, caractérisé en ce qu'il comporte un entrelaceur selon l'une quelconque des revendications 10 à 15 ou un dispositif de codage selon l'une quelconque des revendications 16 ou 17 ou un dispositif de décodage selon la revendications 18.

20. Dispositif de transmission de données comportant un émetteur adapté à mettre en oeuvre un protocole de transmission par paquets, caractérisé en ce qu'il comporte un entrelaceur selon l'une quelconque des revendications 10 à 15 ou un dispositif de codage selon l'une quelconque des revendications 16 ou 17 ou un dispositif de décodage selon la revendications 18 ou un dispositif de traitement de signaux représentatifs de parole selon la revendication 19.

21. Dispositif de transmission de données selon la revendication 20, caractérisé en ce que ledit protocole est le protocole ATM (acronyme des mots anglais "Asynchronous Transfert Mode") mode de transfert asynchrone.

22. Dispositif de transmission de données comportant un émetteur émettant sur un canal non filaire, caractérisé en ce qu'il comporte un entrelaceur selon l'une quelconque des revendications 10 à 15 ou un dispositif de codage selon l'une quelconque des revendications 16 ou 17 ou un dispositif de décodage selon la revendications 18 ou un dispositif de traitement de signaux représentatifs de parole selon la revendication 19 ou un dispositif de transmission de données selon l'une quelconque des revendications 20 ou 21.

23. Dispositif de traitement de séquences de signaux représentatifs d'au plus mille données binaires, caractérisé en ce qu'il comporte un entrelaceur selon l'une quelconque des revendications 10 à 15 ou un dispositif de codage selon l'une quelconque des revendications 16 ou 17 ou un dispositif de décodage selon la revendications 18 ou un dispositif de traitement de signaux représentatifs de parole selon la revendication 19 ou un dispositif de transmission de données selon l'une quelconque des revendications 20 ou 21 ou un dispositif de transmission de données selon la revendication 22.

24. Station de réseau, caractérisé en ce qu'il comporte un entrelaceur selon l'une quelconque des revendications 10 à 15 ou un dispositif de codage selon l'une quelconque des revendications 16 ou 17 ou un dispositif de décodage selon la revendications 18 ou un dispositif de traitement de signaux représentatifs de parole selon la revendication 19 ou un dispositif de transmission de données selon l'une quelconque des revendications 20 ou 21 ou un dispositif de transmission de données selon la revendication 22 ou un dispositif de traitement de séquences de signaux selon la revendication 23.

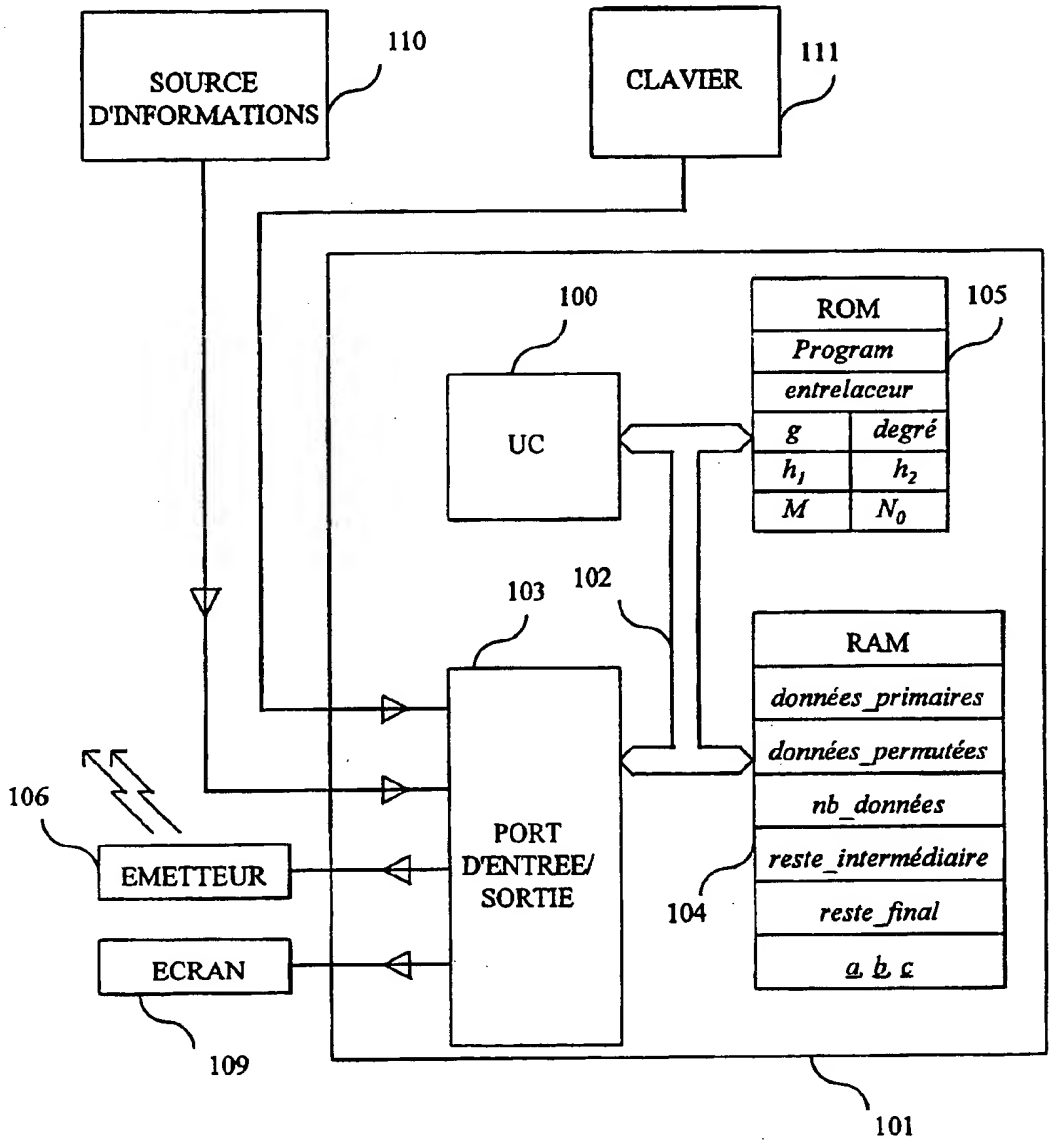

SOURCE D'INFORMATIONS
110
CLAVIER
111
100
UC
ROM
105
Program
entrelaceur
g
degré
h_1
h_2
M
N_0
103
102
RAM
données_primaires
données_permutées
nb_données
reste_intermédiaire
reste_final
a, b, c
104
PORT D'ENTREE/ SORTIE
106
EMETTEUR
ECRAN
109
101

Fig. 1

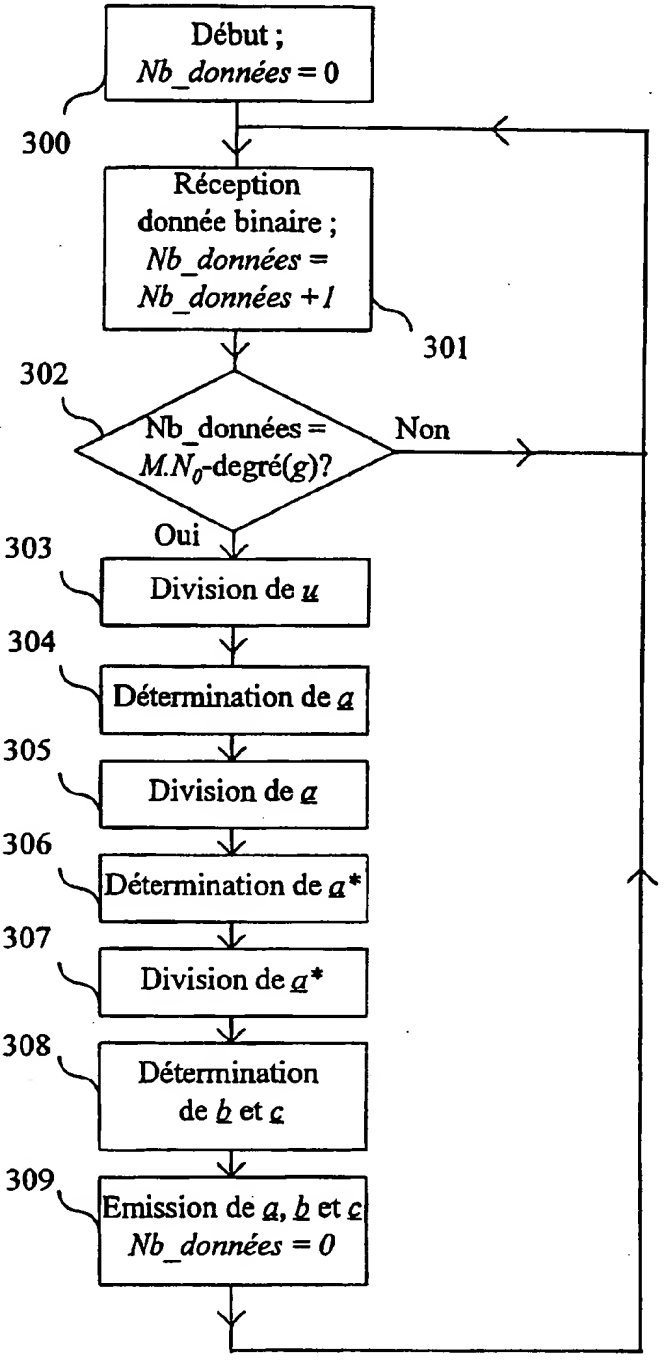

Début ;
Nb_données = 0
300
Réception
donnée binaire ;
Nb_données =
Nb_données +1
301
302
Nb_données =
M.N0-degré(g)?
Non
Oui
303
Division de u
304
Détermination de a
305
Division de a
306
Détermination de a*
307
Division de a*
308
Détermination
de b et c
309
Emission de a, b et c
Nb_données = 0

Fig. 2

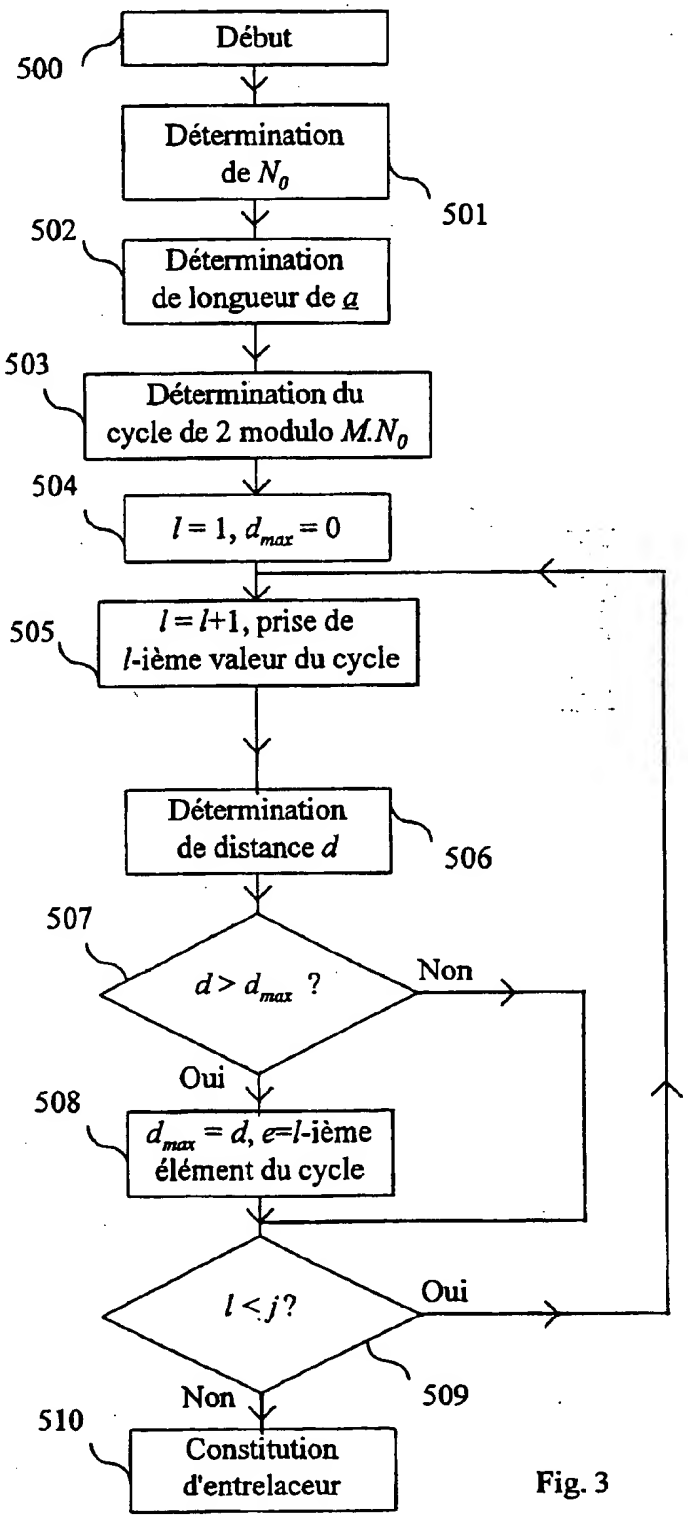

Début
500
Détermination de N_0
501
502
Détermination de longueur de a
503
Détermination du cycle de 2 modulo $M.N_0$
504
$l = 1, d_{max} = 0$
505
$l = l+1$, prise de l-ième valeur du cycle
Détermination de distance d
506
507
$d > d_{max}$?
Non
Oui
508
$d_{max} = d$, e=l-ième élément du cycle
$l < j$?
Oui
509
Non
510
Constitution d'entrelaceur

Fig. 3

REPUBLIQUE FRANÇAISE

2773287

INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE PRELIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

N° d'enregistrement national

FA 553112
FR 9716669

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée	
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
D,A	BERROU ET AL.: "Near Optimum error correcting coding and decoding : turbo codes" IEEE TRANSACTIONS ON COMMUNICATIONS, vol. 44, no. 10, octobre 1996, pages 1261-1271, XP000629465 IEEE Transactions on Communications paragraphes "Uniform interleaving" et "Nonuniform interleaving"	1,2,7, 9-11,16, 18-20, 22-24	
A	DIVSALAR ET AL.: "Turbo codes for PCS applications" PROCEEDINGS OF THE INT'L CONFERENCE ON COMMUNICATIONS, vol. 1, 18 - 22 juin 1995, pages 54-59, XP000532968 Seattle, US paragraphe "Interleaver design"		
D,A	BARBULESCU ET AL.: "Interleaver design for turbo codes" ELECTRONICS LETTERS, vol. 30, no. 25, 8 décembre 1994, pages 2107-2108, XP000501850 Stevenage, Herts , GB * le document en entier *		DOMAINES TECHNIQUES RECHERCHES (Int.Cl.6) H03M
A	BERROU ET AL.: "Frame oriented convolutional turbo codes" ELECTRONICS LETTERS, vol. 32, no. 15, 18 juillet 1996, pages 1362-1364, XP000625424 Stevenage, Herts, GB paragraphe " Theorem 1"		

1

Date d'achèvement de la recherche	Examinateur
6 octobre 1998	Augarde, E

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 03.82 (P04C13)